University of Washington – Computer Science & Engineering

Winter 2017 Instructor: Justin Hsia 2017-03-13

CSE 369 QUIZ 3

Name: _____

UWNetID: _____

Please do not turn the page until 10:30.

Instructions

- This quiz contains 4 pages, including this cover page. You may use the backs of the pages for scratch work.
- Please clearly indicate (box, circle) your final answer.
- The quiz is closed book and closed notes.
- Please silence and put away all cell phones and other mobile or noise-making devices.
- Remove all hats, headphones, and watches.
- You have 35 minutes to complete this quiz.

Advice

- Read questions carefully before starting. Read *all* questions first and start where you feel the most confident to maximize the use of your time.
- There may be partial credit for incomplete answers; please show your work.
- Relax. You are here to learn.

Question	Points	Score
(1) Counters	12	
(2) Shift Registers	11	
(3) Routing Elements	9	
Total:	**32**	

Question 1: Counters [12 pts]

Implement a counter that goes through the following state sequence: **000 → 111 → 110 → 101 → 001** → 000 → ... using a *minimal number of 2-input logic gates.*

PS_2	PS_1	PS_0	NS_2	NS_1	NS_0
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

NS_2

	00	01	11	10
0				
1				

NS_1

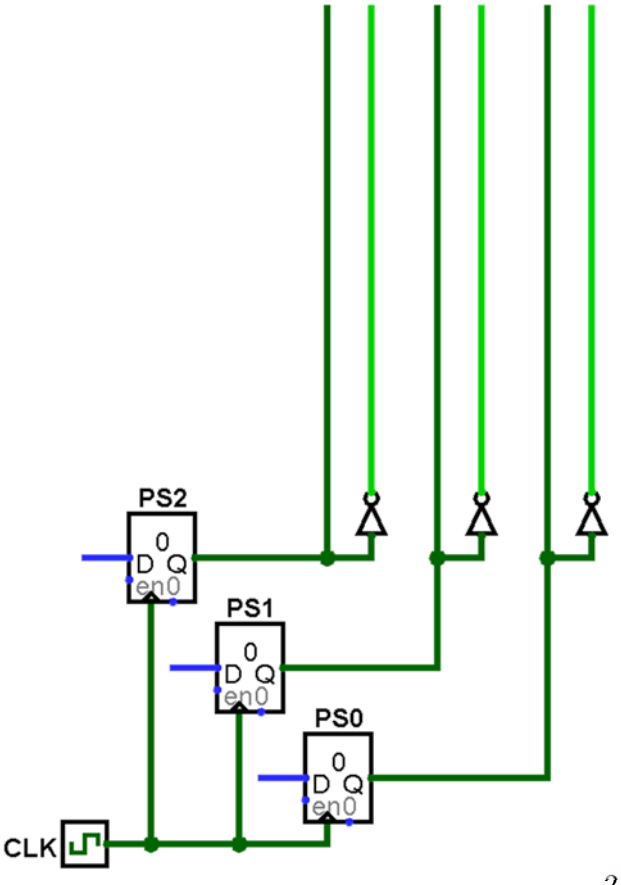
	00	01	11	10
0				
1				

NS_0

	00	01	11	10
0				
1				

Wire connection: 

Wire crossing: 



Question 2: Shift Registers [11 pts]

We have a 4-bit linear feedback shift register (LFSR) that goes through the following state sequence: 0000 → 0001 → 0010 → 0101 → 1011 → ⋯

(A) Circle one: This LFSR is shifting bits to the **LEFT** / **RIGHT**. [1 pt]

(B) The bit that is shifted in is a function of two of the LFSR bits. We number the bits starting from 0 increasing from right to left (like standard binary). What is the name of the gate being used and which two bits are its inputs? [6 pt]

Gate:	
Bits:	_____ and _____

(C) Complete the Verilog code below to implement this LFSR. Feel free to use "**state[i] ? state[j]**" to indicate the correct answer to part B. [4 pt]

```
module LFSR (state, shift, reset, clk);
  output reg [3:0] state;
  input      shift, reset, clk;

  always @(posedge clk) begin

    if ( ______ )

      state <= ____________________________________;

    else if ( ______ )

      state <= ____________________________________;
  end

endmodule
```

Question 3: Routing Elements [9 pts]

Implement a circuit that computes the **factorial function `n! = n*(n-1)!`**. Note that it will take `n` clock cycles to compute `n!` and we will let it run infinitely (*no stop condition*).

Note 1: Both registers (after a `Reset`) start with value 0. Make sure that your circuit doesn't get stuck at the value 0. Hint: what's the title of this problem?

Note 2: Make sure that your `n` and `n!` bus values line up properly (other than 0! and 0).

Assume you can freely use gates and routing elements discussed in class plus the constants **`0`** and **`1`** and the following logic blocks: